Field Session Fall 2021 - Requirements
Project: Medecipher: Nurse Schedule Builder User Interface
Team: Collette Haberland, Patrick Curran, Daniel Ayers, Grant Josenhans

Steve Tinsley: CTO
Kevin Bui: Operation Researcher (Data Science division)
Connor Graflund: Software Engineer

High-Level Description/Vision

Medecipher is striving towards using machine learning to optimize nurse-to-patient ratios in hospitals and prevent nurse burnout. Medecipher has created a scheduling program that forecasts hospital patient count and schedules nurses appropriately. In order to fully utilize the technology Medecipher is developing, a user interface usable by a hospital schedule manager needs to be developed.

Our team will design and create a client facing user interface that allows clients to customize nurse schedules. The UI should receive schedule data from an API, allow the client to view and modify the schedule, and save any modifications made to the schedule.

Functional Requirements

Functional requirements include a UI layer and an API layer. The UI will receive schedule data from the API layer and update modifications to schedule. The UI will also have the ability to make staffing recommendation choices. It will display a schedule view and which user interacts with to choose date and recommendation type and the UI will send requests to REST API which pulls back and caches data, returns data points and schedule that displays recommendations. UI will also provide warnings related to sparsity and nurse burn-out. The API layer abstracts out communication with API. API layer will connect between the UI and the data store and microservices/data science layer that has already been developed. We will also create a continuous delivery pipeline for automated testing. Unit tests may be conducted with API calls and end-to-end tests will be performed similarly except as a user instead of direct API calls.

Non-Functional Requirements

The non-functional requirements include the operating system specified by Medecipher. Linux/Windows will be used for the UI, and Linux for the API. The majority of tasks will be completed locally and will require much processing power. Requirements will mainly be handled through Amazon Web Services. These include the size of clusters, elastic costs, authentication/security, etc. The main requirement revolves around an automated, simple visual that follows standard coding practices.

Risks

There are a number of technical and skill related risks involved in the project. One obvious risk in the field of healthcare is a breach in confidentiality with respect to personal health information. The data we will be working with will be stripped of said information so we will not need to worry about personally encrypting or protecting this information. However, we will need to ensure that the data we are retrieving has indeed been correctly stripped of all private information. We will continue to remain in contact with Medecipher about insecure data if that happens to be the case. We will then take further actions to ensure the data is as secure as possible, under Medecipher's standards.

In terms of skill risks, the greatest risk comes from our team's lack of experience with Angular10 and app design architecture. The risk is that the appearance and user-friendliness of our application will fall short of their potential because of our lack of experience in this area. Beyond the software not being as high quality as we would like, there is also a risk that users will ignore conflict or overwork warnings if we do not implement them in a way that forces the user to address them. If the user could ignore these warnings, it could potentially cause problems for nurses and their employers.

Definition of Done

By the end of the semester, we aim to have a functional user interface that properly processes and updates data. In order to test our product our client will upload a schedule and make changes to it by adding and deleting shifts. Conflicts, warnings, and other notices will provide useful and relevant information to assist the person creating the schedule. The final product will be delivered by granting Medecipher access to the AWS cluster.

Final Thoughts

After our initial meeting with Medecipher we have gained an overall picture of Medecipher's vision for our project. We will continuously communicate with them, via email, to constantly provide updates about our features, requirements, tests, and more. Our team will also attend Medecipher's Friday's tech bi-weekly meetings. Many of the general guidelines for our project are up to us to be creative and as effective as possible.